Express MailNo.: EV 269189261 US

PATENT
APPL-P2815

This application is submitted in the names of Inventors: Jerrold Von Hauck, Prashant Kanhere, William Duckwall, assignors to Apple Computer Inc.

SPECIFICATION

FLY-BY AND ACK-ACCELERATED ARBITRATION FOR BROADCAST PACKETS

BACKGROUND OF THE INVENTION

[0001] A "bus" is a collection of signals interconnecting two or more electrical devices that permits one device to transmit information to one or more other devices. There are many different types of busses used in computers and computer-related products. Examples include the Peripheral Component Interconnect ("PCI") bus, the Industry Standard Architecture ("ISA") bus and the Universal Serial Bus ("USB"), to name a few. Bus operation is usually defined by a standard that specifies various concerns such as the electrical characteristics of the bus, how data is to be transmitted over the bus, how requests for data are acknowledged, and the like. Using a bus to perform an activity, such as transmitting data, requesting data, etc., is generally called running a "cycle." Standardizing a bus protocol helps to ensure effective communication between devices connected to the bus, even if such devices are made by different manufacturers. Any company wishing to make and sell a device to be used on a particular bus, provides that device with an interface unique to the bus to which the device will

connect. Designing a device to particular bus standard ensures that device will be able to communicate properly with all other devices connected to the same bus, even if such other devices are made by different manufacturers.

[0002] Thus, for example, an internal fax/modem (i.e., internal to a personal computer) designed for operation on a PCI bus will be able to transmit and receive data to and from other devices on the PCI bus, even if each device on the PCI bus is made by a different manufacturer.

[0003] According to most bus protocols, a device that needs to run a cycle on the bus must first gain control of the bus. Once the sending device has control of the bus, that device then can run its desired cycle, which may entail transmitting data to a receiving bus device. Often, more than one bus device may concurrently need to initiate a cycle on the bus. Bus protocols in which multiple devices may request control of the bus to run cycles usually implement some form of "arbitration" to efficiently decide which device to grant control of the bus among multiple devices requesting control. The prior art is replete with many types of arbitration schemes.

[0004] Currently, there is a market push to incorporate various types of consumer electronic equipment with a bus interface that permits such equipment to be connected to other equipment with a corresponding bus interface. For example, digital cameras, digital video recorders, digital video disks ("DVDs"), printers are becoming available with an IEEE 1394 bus interface. The IEEE ("Institute of Electrical and Electronics Engineers") 1394 serial interface (and all its variations, referred to collectively herein as "1394") describes a bus that permits a digital camera to be connected to a printer or computer so that an image acquired by the camera can be printed on the printer or stored electronically in the computer. Further, digital televisions can be coupled to a computer or computer network via an IEEE 1394 bus.

[0005] Asynchronous stream packets are a form of packet incorporated in 1394 to circumvent older link design packet-filtering limitations. Asynchronous stream packets are like isochronous packets, in that they share tcode "A," and the channel number is allocated in the usual way from the CHANNELS_ AVAILABLE register of the isochronous resource manager.

[0006] However, asynchronous stream packets are transmitted during an asynchronous period, and are subject to the same arbitration requirements as other asynchronous packets (i.e. fairness). Like isochronous packets, there is no ack generated in response. Thus they behave like broadcast packets.

[0007] Existing links filter asynchronous stream packets out successfully in hardware. This is not the case for true broadcast packets, which have a destination ID of 63. Broadcast packets are passed upwards for software to filter, which causes overflows in some implementations.

[0008] Asynchronous streams are heavily used for internet protocol over 1394-compliant systems; on some 1394 systems, a sizable portion of the packets sent could be asynchronous stream packets. Because they are asynchronous packets, without following acknowledge packets, the bus is forced to wait for a full subaction gap before arbitration can begin for the next packet. This effectively drops bus efficiency back to pre-1394a-2000 levels. Ack acceleration, fly-by concatenations, link concatenations are all precluded by an asynchronous stream packet. After an asynchronous stream packet is transmitted on the bus, all nodes connected to the bus must wait for at least a subaction gap time period to pass before beginning bus arbitration.

SUMMARY OF THE INVENTION

[0009] The present invention solves the problems discussed above. In embodiments of the present invention, bus arbitration and/or packet transmission begins much sooner, without waiting for a gap timeout, thus greatly reducing the idle time on the 1394-compliant bus. This increases the effective bus bandwidth. While the present invention is described below with respect to asynchronous stream packets, it is to be understood by those skilled in the art that, unless otherwise stated, the present invention can be used with any asynchronous packets that do not require an unarbitrated response or ack packet sent in response to transmission of the packets, such as asynchronous broadcast packets, some PHY layer packets, and the like.

[0010] In an embodiment, the present invention provides a method for administering a serial bus, the bus facilitating communication between node devices connected to the bus and communicating over the bus in the form of packetized communication between node devices, the method comprising the acts of sending a stream of asynchronous stream packets; if there is an asynchronous packet to be sent, then concatenating the asynchronous packet to the asynchronous stream packets to be sent and sending the asynchronous stream packets followed by the asynchronous packet; and if there is no asynchronous packet to be sent, then concatenating a bogus ack packet to the asynchronous stream packets and sending the asynchronous stream packets and the concatenated bogus ack packet.

[0011] In another embodiment, the present invention provides a method for administering a serial bus, the bus facilitating communication between node devices connected to the bus and communicating over the bus in the form of packetized communication between node devices, the method comprising the acts of receiving an asynchronous stream packet; if there are more asynchronous stream packets to be sent, and if fly-by concatenation is permitted, then concatenate asynchronous stream packets;

and if there are more asynchronous stream packets to be sent and fly-by concatenation is not permitted then arbitrate for control of the bus and sending packets.

[0012] In yet another embodiment, the present invention provides a method for administering a serial bus, the bus facilitating communication between node devices connected to the bus and communicating over the bus in the form of packetized communication between node devices, the method comprising the acts of receiving an asynchronous stream packet; if there is an asynchronous packet to be sent, and if fly-by concatenation is permitted, then concatenate the asynchronous packet to the asynchronous stream packet; if there is an asynchronous packet to be sent and fly-by concatentation is not permitted then arbitrate for the bus; and sending packets.

[0013] In yet another embodiment, the present invention provides a method for administering a serial bus, the bus facilitating communication between node devices connected to the bus and communicating over the bus in the form of packetized communication between node devices, the method comprising the acts of receiving an asynchronous stream packet; determining that there are no more asynchronous stream packets to be sent; determining that there are no asynchronous packets to be sent; if fly-by concatenation is permitted then concatenating a bogus ack packet; if fly-by concatenation is not permitted then arbitrating for the bus; and sending packets.

[0014] Other features and advantages of the present invention will become apparent to those skilled in the art upon reading the following detailed description and figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] FIG. 1 is a high-level diagram in block diagram form of node devices connected to a high speed serial bus;

[0016] FIG. 2 is a high-level block diagram in block diagram form of a single node shown in FIG. 1;

[0017] FIG. 3 is a diagram in flowchart form of acts performed by a node that is an originator of an asynchronous packet; and

[0018] FIG. 4 is a diagram in flowchart form of acts performed by a node that receives an asynchronous packet.

DETAILED DESCRIPTION OF THE INVENTION

[0019] The present invention provides an improvement to arbitration on the IEEE 1394 bus, although the principles of the present invention may extend to other bus protocols, certainly other bus protocols experiencing a similar arbitration problem to that described below. To understand the nature of the problem solved by the present invention, the general structure and operation of an IEEE 1394 bus and arbitration scheme will now be provided. Referring to FIG. 1, an IEEE 1394 network 50 comprises one or more "nodes," node 1-node 7. A node represents an electronic device(s) with an IEEE 1394 bus interface. A node device may comprise a computer, a digital camera, a digital video recorder, a DVD player, or another type of device having a suitable bus interface. Each node couples to at least one other node. As shown in the exemplary architecture of FIG. 1, node 1 couples both to nodes 2 and 3. Node 3, in turn, couples to nodes 4, and 5 and node 5 also couples to nodes 6 and 7. In general, each node can transmit data to any other node in the network. For example, node 7 can transmit data to node 2, but the transmitted data will pass from node 7 to node 5 to node 3 to node 1 and then, to node 2. Each node contains hardware that implements physical layer services (referred to herein as a "phy" and denoted by the letter "P" in each illustrated node) and hardware that implements link layer services (referred to herein as a "link" and denoted by the letter "L" in each illustrated node).

[0020] Referring to FIG. 2, node 100 generally comprises three "layers": transaction layer 102, link layer 104 and physical layer 106 (referred to as a "PHY"). Transaction layer 102 implements the request-response protocol required to conform to the IEEE 1394 standard in accordance with known techniques. Link layer 104 supplies an acknowledgment to the transaction layer. Link layer 104 handles all packet transmission and reception responsibilities as well as the provision of cycle control for isochronous channels. The PHY 106 generally provides the initialization and arbitration services necessary to assure that only one node at a time is sending data and to translate the serial bus data stream and signal levels to those required by the link layer logic 104. PHY 106 also implements the arbitration scheme of the preferred embodiment of the invention.

[0021] Referring still to FIG. 2, PHY 106 preferably includes PHY-Link interface 108, port controller 110, packet transmit/receive 112, BOSS arbitration and control state machine 114, one or more port logic units 116, and physical media dependent ("PMD") electronics unit 118 for each port 116. Each port 116 can be used to couple the nodes to another node in the network. Multiple ports can be included within each node. Although two ports 116 are shown in the preferred embodiment of FIG. 2, the number of ports is not important for the present invention. Each port 116 couples to a PMD 118 that provides the necessary electrical interface to the particular physical communications medium. The physical communications medium may include any suitable type of medium such as Cat. 5 UTP, glass optical fiber, plastic optical fiber, beta-only electrical, bilingual electrical or DS-only electrical types of communication media. Each port 116 and its associated PMD 118 can be connected via a communication medium to another node in the network.

[0022] Port controller 110 generally controls the interface to another node in the network. Packet transmit/receive 112 generally receives and forwards all data packets. Packet transmit/receive 112 includes logic (not shown) to effectively control the flow of data cycles through PHY 106. Any node in the network can request ownership of the network to be the BOSS by transmitting an appropriate request to all of the neighboring

nodes to which the requesting node connects. As such, each node in the network can receive requests from its neighboring nodes to be the BOSS and each node itself can request ownership of the bus. BOSS arbitration and control state machine 114 within a node receives all of the requests from its neighboring nodes in the network, via the various port logic units 116 and PMDs 118, as well as its own request to be BOSS. BOSS arbitration and control state machine 114 then prioritizes these various requests and sends out the highest priority request to all neighboring nodes through all active ports 116 not currently being used to transmit data. Eventually, the requests propagate their way to the current BOSS who grants ownership of the bus to the highest priority requests. Generally, IEEE 1394 supports two types of data transfers: asynchronous and isochronous. Asynchronous data transfers are sent to a specific address and an acknowledgment is returned to verify receipt of the data. Isochronous transfers guarantee data transfers at a predetermined rate that is particularly beneficial for time-critical, multimedia data transfers. Arbitration decisions are made separately for the asynchronous requests and isochronous requests. Accordingly, BOSS arbitration and control state machine 114 prioritizes the asynchronous requests separately from the isochronous requests and provides the highest priority request from each type (asynchronous and isochronous) to the ports not currently transmitting data for transmission to other nodes. The arbitration scheme described above is one style of arbitration used by 1394 PHY layers and is provided for illustrative purposes. Other arbitration schemes for variations on the 1394 standard have slight variations.

[0023] While the present invention is described below with reference to the IEEE 1394 protocols, the present invention in its various embodiments may also be utilized with asynchronous packet transmission performed in accordance with the discovery and enumeration protocol (DEP).

[0024] The present invention can be explained with reference to FIG. 1. The link that sends the asynchronous stream packet concatenates another packet onto the tail of the asynchronous stream packet. In an embodiment, the concatenated packet can be a cycle-start packet, or another asynchronous stream packet, or a normal asynchronous packet, or a bogus ack packet. An isochronous packet should not be concatenated onto

an asynchronous stream packet, because by definition the asynchronous stream packet was transmitted during the asynchronous period of bus operation.

[0025] In the particular case where the link concatenates a bogus ack packet onto the asynchronous stream packet – the bogus ack packet serves to trigger the arbitration acceleration mechanisms available in 1394. Thus, other nodes can employ ack-accelerated arbitration and/or fly-by concatenation after passage of the bogus ack packet.

[0026] The bogus ack packet does not have to follow the prior-art convention that the second nibble is the complement of the first nibble. By not having a second nibble that is the compliment of the first nibble, the bogus ack packet causes the link hardware to filter it out so that it has no effect at that level. However, by definition in 1394, the PHY recognizes any 8-bit packet as an ack packet for the purposes of acceleration. Thus the present invention utilizes a packet that meets the requirements for recognition at one physical and/or transaction layer, but not at a different layer.

[0027] In an embodiment, originator concatenation is performed by link hardware. In embodiments, fly-by concatenation, fly-by concatenation of bogus ack packet, early arbitration and early arbitration for bogus ack packet are performed by PHY hardware. In an embodiment, originator bogus ack packet generation may be performed by both link hardware and PHY hardware. If the link performs the function, then the PHY defers, even if the node incorporates a PHY without the functionality of the present invention. Similarly, a PHY with the functionality of the present invention is able to concatenate the bogus ack packet, even if the link hardware has no knowledge of the action.

[0028] For accelerated bus arbitration upon detection of an asynchronous stream packet, the PHY saves the first quadlet of every passing packet, so that the tcode field can be checked. The PHY keeps track of whether it is in an isochronous period or not so that it doesn't start sending fair/pri packets during an isochronous period. In an embodiment, the start of an isochronous period can be by detection of a cycle start (where tcode =8).

The end of the isochronous period can be detected at the occurrence of the first subaction (SA) gap.

[0029] In an embodiment, to identify other asynchronous packet types that do not require an unarbitrated response or ack packet sent in response to the transmission of the packet, the PHY can inspect the first quadlet of passing packets to determine the packet type.

[0030] Automatic sending of a bogus ack packet can be included in an embodiment of the present invention for the purpose of enabling ack accelerated arbitration after an asynchronous packet stream by an entity other than the sender of the asynchronous stream packet, e.g. the root node. One enlightened node on bus 50 can accelerate arbitration for the entire bus, even if populated by less enlightened hardware. A node can keep track of the isochronous period to keep from generating acks for isochronous packets. This approach works even if multiple nodes attempted to send the bogus ack packet, provided they arbitrate to send it rather than send it as an immediate packet or fly-by concatenating it). That way, one node wins arbitration and sends the bogus ack, and others get back to accelerated arbitrations after detecting an ack packet.

[0031] FIG. 3 illustrates in flow chart form a sequence of steps utilized in embodiments of the present invention by the originator of an asynchronous stream packet. Directing attention to acts 200 and 202, asynchronous stream packets are sent or concatenated to other asynchronous stream packets. At act 204, if there is an asynchronous packet to send after the asynchronous stream packets have been sent or concatenated, control transitions to act 206, where the asynchronous packet is concatenated and sent. However, if there is no asynchronous packet to send, a bogus ack packet is concatenated and sent at act 208.

[0032] FIG. 4 illustrates in flow chart sequences of acts performed by the receiver of an asynchronous stream packet. At act 300, an asynchronous stream packet is received. If there are more asynchronous stream packets to be sent (decision act 302), control

proceeds to decision act 304, where a determination is made as to whether fly-by concatenation is permitted. If fly-by concatenation is permitted, control proceeds to act 308, where the asynchronous stream packet is concatenated and sent. Control then returns to act 302. If fly-by concatenation is not permitted, control transitions from act 304 to act 306, where bus 50 is arbitrated. At decision act 310, a determination is made as to whether there is an asynchronous packet to send. If there is an asynchronous packet to send, control proceeds to decision act 312, where a determination is made as to whether fly-by concatenation is permitted. If fly-by concatenation is permitted, control proceeds to act 314, where the asynchronous packet is concatenated and sent. However, if fly-by concatenation is not permitted, control proceeds to act 316, where bus 50 is arbitrated. Returning to decision act 310, if there is not an asynchronous packet to send, control transitions to decision act 318, where a determination is made as to whether fly-by concatenation is permitted. If fly-by concatenation is permitted, control proceeds to act 320, where a bogus ack packet is concatenated and sent. If fly-by concatenation is not permitted, control proceeds to act 322, where bus 50 is arbitrated, then a bogus ack packet is sent at act 324.

[0033] While various methodologies for administering a high-speed serial bus have been described and illustrated above, it is to be understood that various modifications can be made to embodiments of the present invention without departing from the spirit thereof.